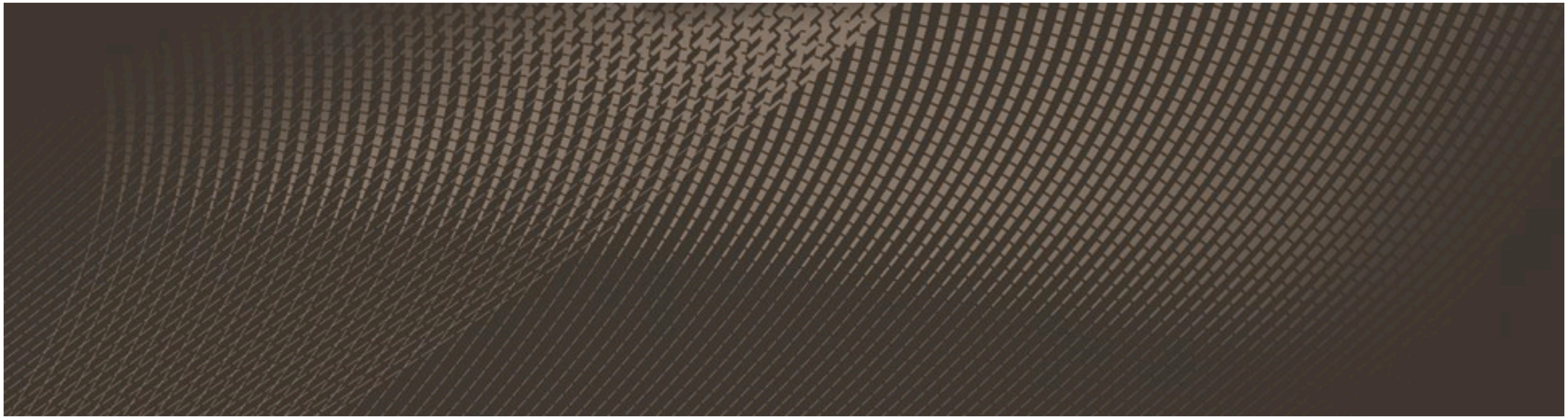

The good, the bad and the un-debuggable

A quick look at HTML5 Caching and localStorage

Andreas Heim - May 23 2011 – Roots Conference, Bergen

BEKK

The good, the bad, and the un-debuggable

HTML5 Caching and localStorage

@heim github.com/heim

why cache?

load time

responsive webapp

possibility for offline use

lots of files equals
lots of http-overhead

reduce server load

better UX

a good strategy:

load only static assets

(css, images, js)

populate with data over lightweight protocol (json) when navigating, replace data instead of reloading page

result:
a snappy webpage

a better strategy:

load and cache static assets

load, populate and cache data via lightweight protocol show old data and load new data in the background

prefer stale data

cached assets are loaded once from the server

CACHE MANIFEST

define which pages
and assets that
should be cached

HTML5

“I’ll look at it when I can use it…”

“...like when the kids are grown up...”

“...and wasn’t the world supposed to end last saturday?”

compatibility

- ~~IE: no~~
- Firefox 3.5+
- Chrome 5.0+
- Safari 4.0+
- Mobile Safari 2.1+
- Android 2.0+

time to grow up!

and by the way

both users of Opera 10.6+

one line in your html

```
<html manifest=”application.manifest”>
```

has to be served with correct content type

content-type=”text/cache-manifest”

```
CACHE MANIFEST
/application.css
/application.js
/images/logo.png
```

all pages referring to manifest is implicitly part of it no need to cache it all eh?

```
CACHE MANIFEST

CACHE:
/application.css

NETWORK:
/tracking.cgi
```

fallback pages for when the connection drops

```
CACHE MANIFEST

CACHE:
/application.css

FALLBACK:
/ /offline.html
```

assets are only reloaded when the manifest file changes then every single resource is reloaded and cached

the event flow of the cache

browser finds new manifest

fires event “downloading” and downloads files in the background

fires event “progress” at seemingly random intervals and with random data

fires “cached” when done

browser finds known manifest

if the manifest file itself hasn't changed it does nothing and fires event "noupdate"

manifest is known and has changed

fires “downloading”

fires “progress”

fires “updateready” when done

the bad

what could possibly go wrong with this setup?

the manifest file could be ill-formed

a resource refered to from the manifest file can be inaccessible the manifest was updated while someone was downloading one of the assets one of the files in the manifest does not exist something on the cached page fails

if any of this happens

everything fails

silently

please, put this in
your js

```
$(window.applicationCache).bind(“error”, function(e) {
    //alert or something
});
```

the ugly

after updating the manifest you have to reload browser twice to get new data fix it like this:

```
$(window.applicationCache).bind(“updateready”, function(e) {
  //reload page
});
```

“i’m in ur localstore”

not a part of html5

Web Storage/DOM Storage

key/value-database

like a giant 5MB cookie

but data exists only
on the client

IE: 8.0+
FF: 3.5+
Safari: 4.0+
Chrome: 4.0+
Opera: 10.5+
iOS: 2.0+
Android:2.0+

lets us store data on the client for better performance like this:

```
localStorage[“key”] = “value”
```

```
var v = localStorage[“key”]
```

```
delete localStorage[“key”]
```

not very exciting

but this is exciting

github.com/wycats/jquery.offline

```
$.retrieveJSON(“items.json”,
   dataToSend,  function(data) {
   //populate html-page
});
```

similar to
jquery.getJSON

but stores data in localStorage

first time a resource is retrieved, the data is cached in localstorage

and the json-payload
is returned

when retrieving a
known resource
it first serves the
cached data

and retrieves fresh data from the server in the background when fresh data from the server has arrived, the callback is executed again

summary

minify http-overhead

app should work while making connection

load data only once

the cache manifest
will make you mad

but your users will thank you

thanks

spkr8.com/heim

@heim

ANDREAS HEIM
CONSULTANT
+47 959 39 833
andreas.heim@bekk.no

BEKK CONSULTING AS
SKUR 39, VIPPETANGEN. P.O. BOX 134 SENTRUM, 0102 OSLO, NORWAY. WWW.BEKK.NO